



FHWS

iOS Programmierung

(mit Swift)

Peter Braun, Florian Bachmann & Andreas Wittmann

[@pe_braun](#)

[@florianbachmann](#)

[@anwittmann](#)

Deutsche Telekom AG

FHWS - Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt

#FHWSSwift

Agenda

1. **Introduction** – Organisatorisches
2. **First iOS-Project** – Hello World, **First iOS-Project** – Still Hello World (now with Code 😊)
3. **Swift**, Wait!, What about Objective-C?, Why Swift?
4. **A (not so) Quick Tour**
5. **Documentation**
6. **The basics** – iOS Architecture & more
7. **User Interfaces** – View Controller, Auto Layout & Size Classes
8. **Storyboard & Segues**
9. **Tables & UINavigationController**
10. **TabBarController**
11. **Notifications**
12. **PickerViews**
13. **Touches, Gestures, 3D Touch, Peek & Pop**
14. **ScrollView & StackViews**
15. **Networking** – JSON & Dependency Managers
16. **WebKit**
17. **Maps**
18. **Storage & Data persistency** – UserDefaults, NSKeyedArchiver & Core Data
19. **ObjC**

Notifications

NSNotificationCenter, Notifications & Observers

Notifications

don't mix up with User
Notifications, these are for the
programmer :-)

NSNotificationCenter

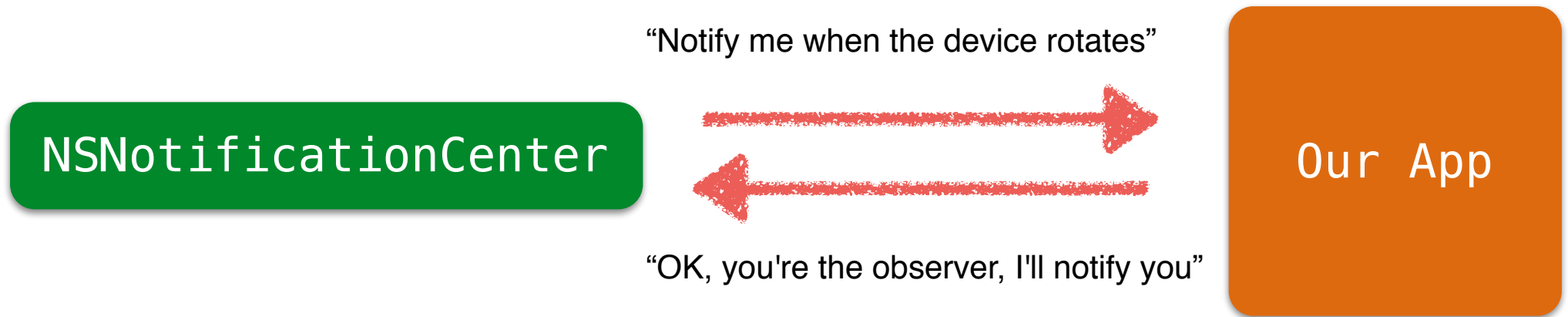
- “The app has become active”
- “A WiFi network has been found”
- “The device has been rotated”
- “The app will quit”
- “The device has changed location”
- “The app is out of memory”

This one everybody has seen:

```
func application(
    application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
    // Override point for customization after application launch.
    return true
}
```

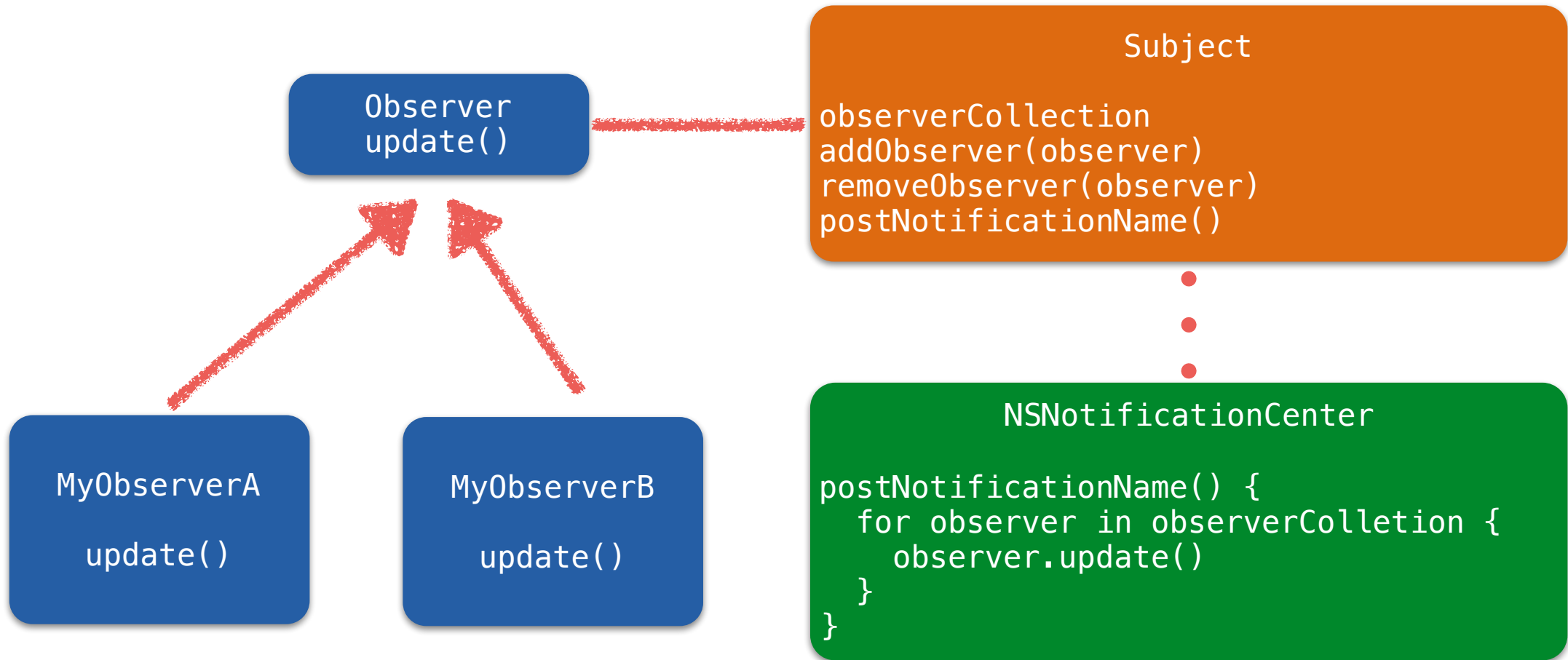
Observers

- Objects that react to events when notified



Observer Pattern

- A common software design pattern



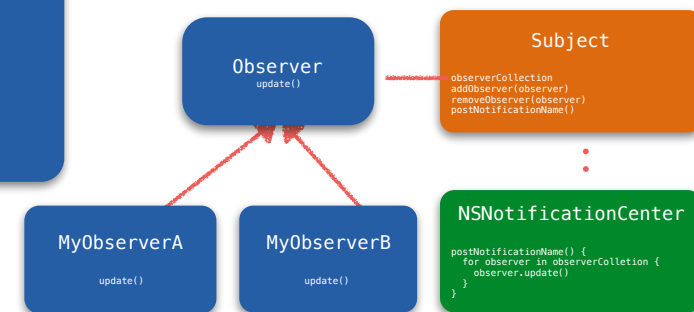
Observer Re-/Deregistration

Telling NotificationCenter when to notify an object

```
let n = NotificationCenter.defaultCenter()
n.addObserver(theObservingObjectThatWantsToBeNotified,
  selector: methodNameToCallOnTheObservingObject,
  name: theNameOfTheNotification,
  object: theSpecificObjectToObserve)
```

Good people deregister the observer, if not needed anymore (dealloc, viewDidLoad)

```
let n = NotificationCenter.defaultCenter()
n.removeObserver(observer)
// alternative:
//Remove all notifications from being observed
n.removeObserver(self)
```



Post Notifications

Telling NotificationCenter when to notify an object

```
let n = NotificationCenter.defaultCenter()  
d.postNotificationName("theNameOfTheNotification", object: nil)
```

And receive them

```
func methodOfReceivedNotification(notification: NSNotification){  
    //Take Action on Notification  
}
```

